Digital System Design

Lecture 10

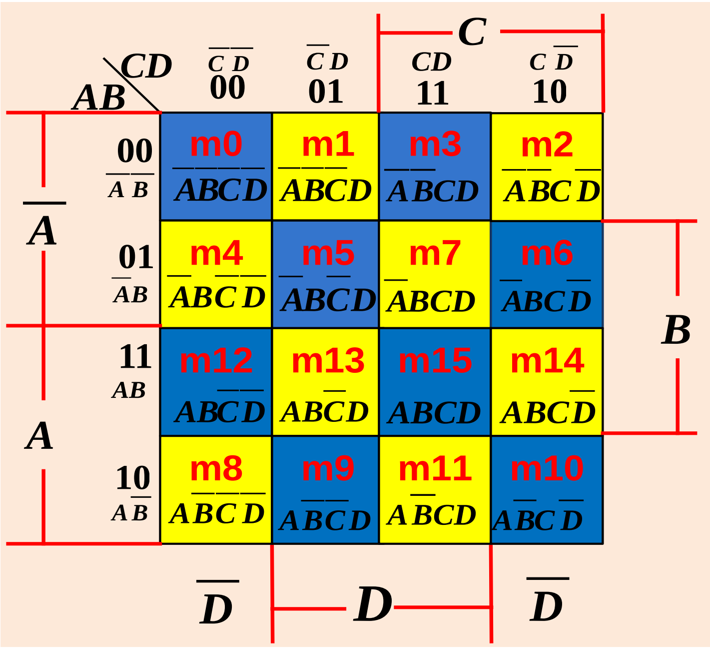
Karnaugh Map Methods- II

Objectives:

1. Four variables maps.
2. Simplification using prime implicants.
3. "Don’t care" conditions.
4. Summary.

1. Four variables Karnaugh maps

A	B	C	D	Minterms
0	0	0	0	$\bar{A}\bar{B}\bar{C}\bar{D}$
0	0	0	1	$\bar{A}\bar{B}\bar{C}D$
0	0	1	0	$\bar{A}\bar{B}C\bar{D}$
0	0	1	1	$\bar{A}\bar{B}CD$
0	1	0	0	$\bar{A}B\bar{C}\bar{D}$
0	1	0	1	$\bar{A}B\bar{C}D$
0	1	1	0	$\bar{A}BC\bar{D}$
0	1	1	1	$\bar{A}BCD$
1	0	0	0	$A\bar{B}\bar{C}\bar{D}$
1	0	0	1	$A\bar{B}\bar{C}D$
1	0	1	0	$A\bar{B}C\bar{D}$
1	0	1	1	$A\bar{B}CD$
1	1	0	0	$AB\bar{C}\bar{D}$
1	1	0	1	$AB\bar{C}D$
1	1	1	0	$ABC\bar{D}$
1	1	1	1	$ABCD$



CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

CD \ AB	00	01	11	10
00	A′B′C′D′	A′B C′D′	A B C′D′	A B′C′D′
01	A′B′C′D	A′B C′D	A B C′D	A B′C′D
11	A′B′C D	A′B C D	A B C D	A B′C D
10	A′B′C D′	A′B C D′	A B C D′	A B′C D′

➢ The rows and columns are numbered in a ***gray code sequence***.

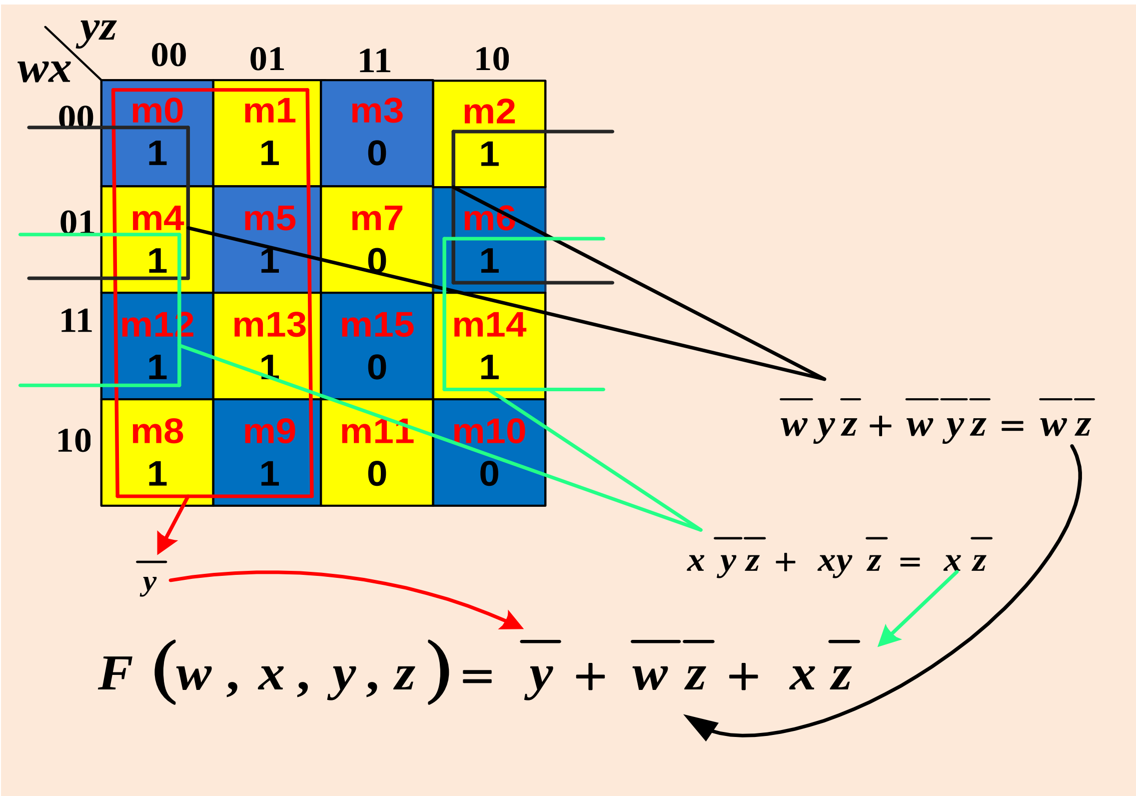
> ✓ One square represents one minterm with four literals.
>
> ✓ Two adjacent squares represent one term with 3 literals.
>
> ✓ Four adjacent squares represent one term with 2 literals.
>
> ✓ Eight adjacent squares represent one term with 1 literal.

Examples:

Example 1: ***Simplify the Boolean function***

$$F(w,x,y,z) = \sum(0,1,2,4,5,6,8,9,12,13,14)$$

Solution:



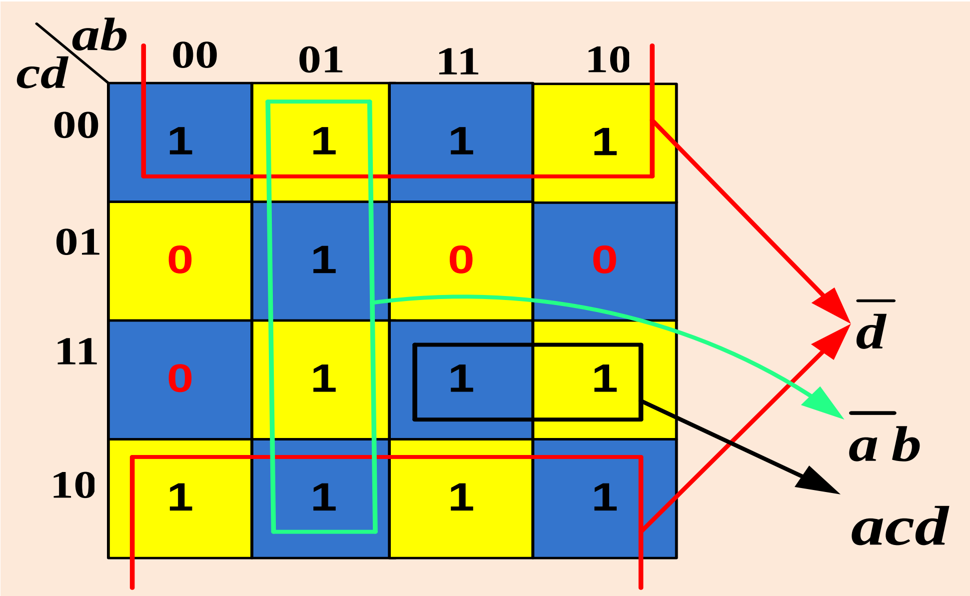
The simplified function:

$$F(w,x,y,z) = \bar{y} + \bar{w}\bar{z} + x\bar{z}$$

Example 2: ***plot the following 4-variable expression on a Karnaugh map***

$$f(a,b,c,d) = acd + \bar{a}b + \bar{d}$$

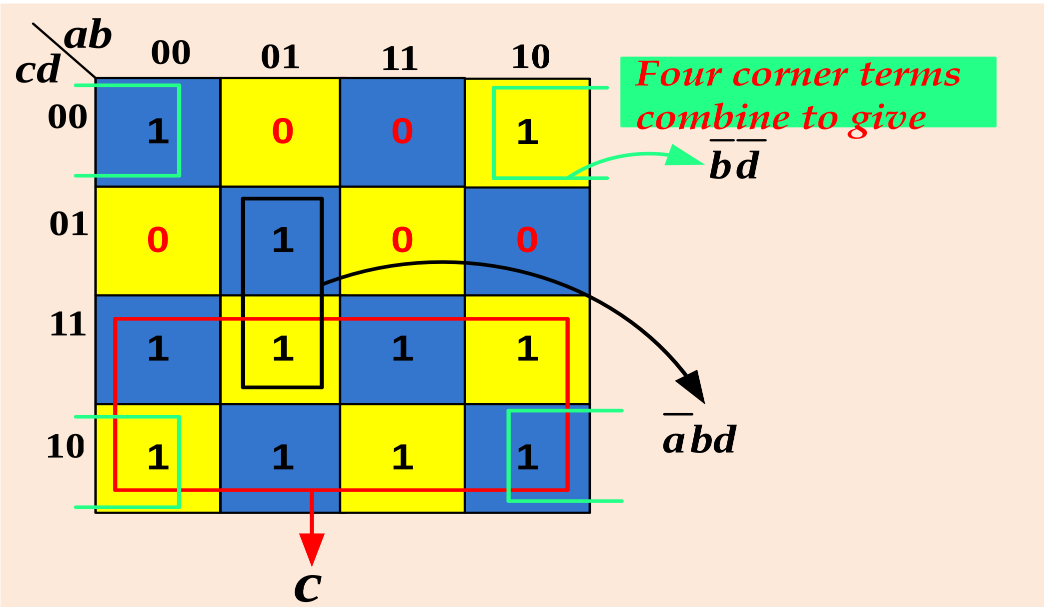
Solution:



Example 3: ***simplify the following function:***

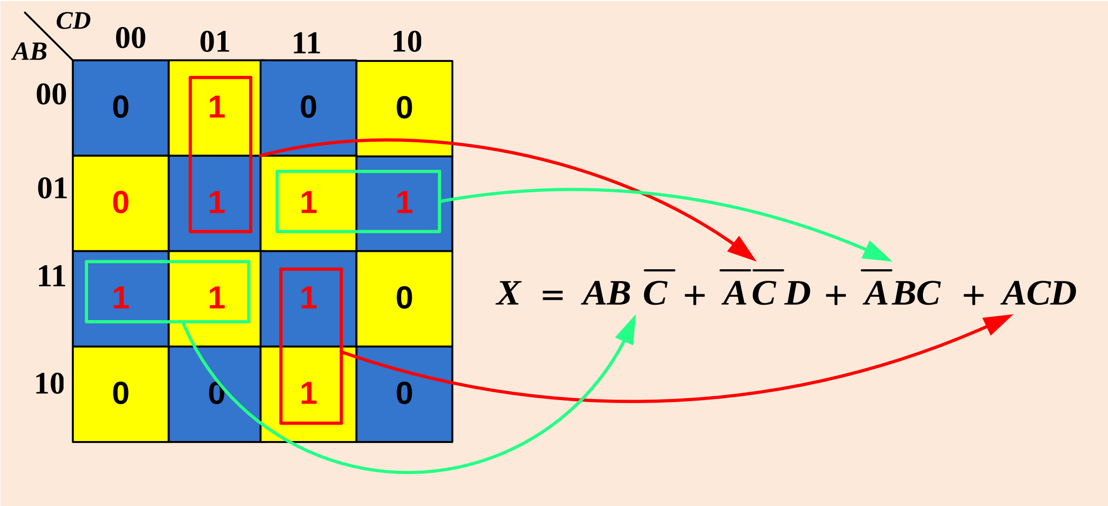
$$f(a,b,c,d) = \sum m(0,2,3,5,6,7,8,10,11,14,15)$$

Solution:



Solution: $f(a,b,c,d) = c + \bar{b}\,\bar{d} + \bar{a}\,bd$

***Example 4:* For the following k-map with four variables, obtain the simplified logic expression:**



***Example 5:* Use a k-map to simplify**

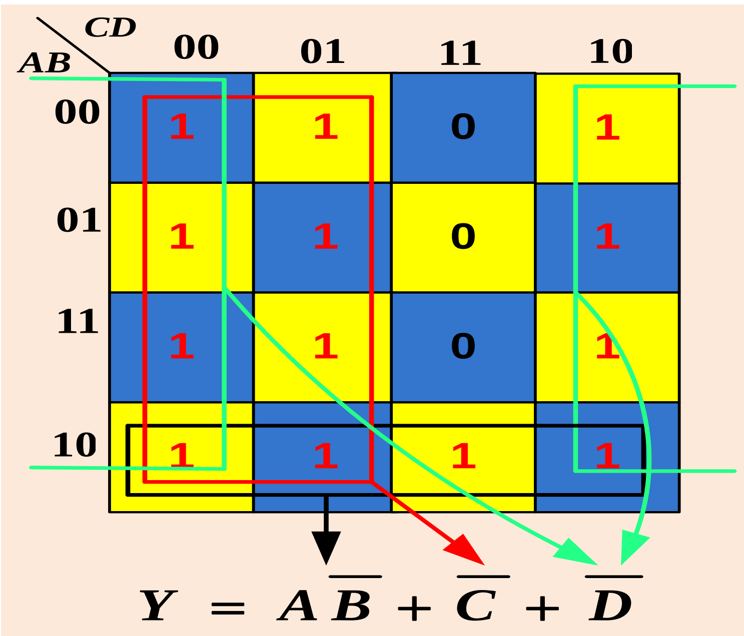
$$Y = \overline{C}(\overline{A}\overline{B}\overline{D} + D) + A\overline{B}C + \overline{D}$$

Solution:

- *Multiply out*

$$Y = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{C}D + A\overline{B}C + \overline{D}$$

- *Fill the terms in k-map:*



- *Simplify:*

$$Y = A\overline{B} + \overline{C} + \overline{D}$$

2. Simplification using prime implicants

Definitions:

- ✓ **Prime implicant (PI)**: is a product term obtained by ***combining the maximum possible number of adjacent squares in the map***.
- ✓ **Essential prime implicant**: If a minterm in a square is ***covered by only one prime implicant*** that prime implicant is said to be ***essential***, and it ***must be included*** in the final expression.

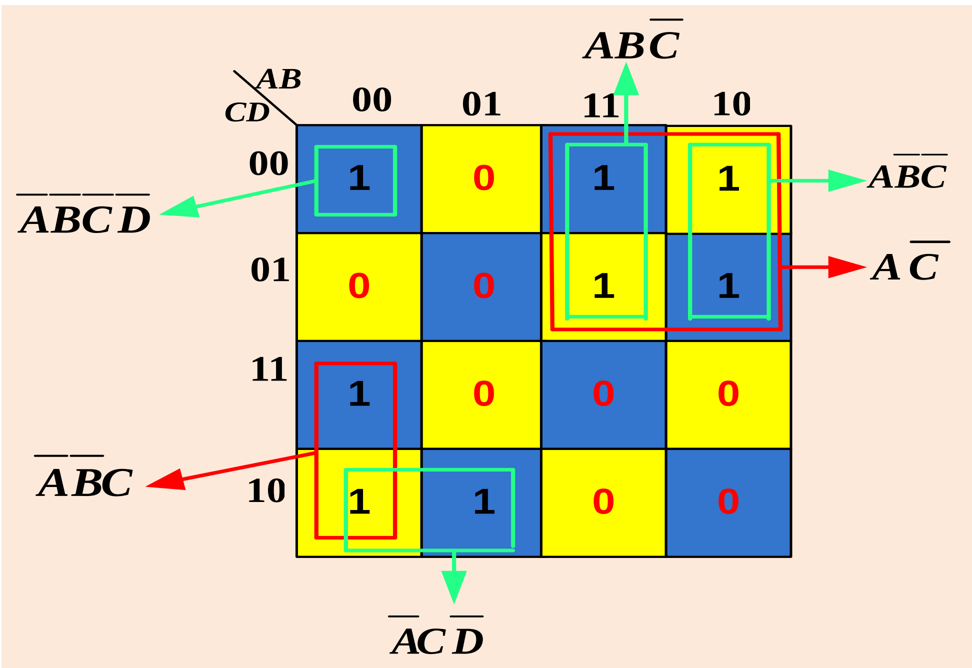
Note:

All of the prime implicants of a function are generally not needed in forming the minimum sum of products.

Procedure for selecting implicants:

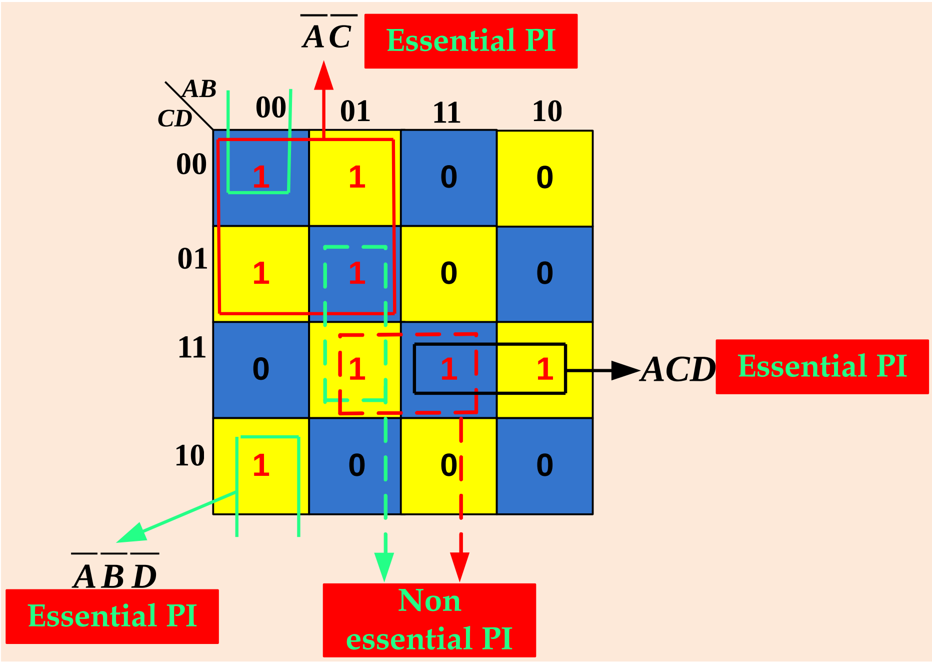
1. Find **essential** prime implicants.
2. Find a minimum set of prime implicants which cover the remaining ***1's*** on the map.

Example 1:



- ✓ $\overline{A}\overline{B}C, \overline{A}C\overline{D}, A\overline{C}$ ***are prime implicants.***
- ✓ $\overline{A}\overline{B}\overline{C}\overline{D}, A\overline{B}\overline{C}, AB\overline{C}$ ***are not prime implicants.***

***Example 2**: find the minimum solution for the following Karnaugh map.*



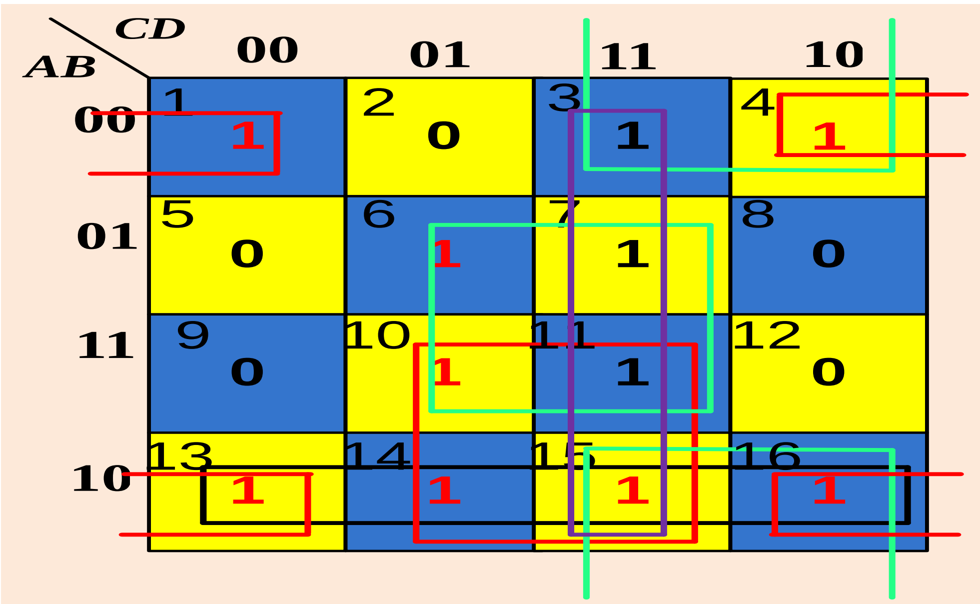
- ✓ $\overline{A}\overline{C}, ACD, \overline{A}\overline{B}\overline{D}$ ***are essential prime implicants, to complete the minimum solution, one of the non-essential prime implicants in needed:***
- ✓ ***The final solution:***

$$F = \overline{A}\overline{C} + \overline{A}\overline{B}\overline{D} + ACD + \{\overline{A}BD \;|BCD\}$$

Example 3:

$$F(A,B,C,D) = \sum m(0,2,3,5,7,8,9,10,11,13,15)$$

Simply using k-map:



Groups (terms number)	**Implicants**	**Simplification**
1, 4, 13, 16	Essential prime implicant	$\overline{B}\overline{D}$
6, 7, 10, 11	Essential prime implicant	BD
3, 4, 15, 16	Prime implicant	$\overline{B}C$
3, 7, 11, 15	Prime implicant	CD
10, 11, 14, 15	Prime implicant	AD
13, 14, 15, 16	Prime implicant	$A\overline{B}$
2 essentials and four prime implicants		

- ✓ ***Square 3*** can be covered with either prime implicants CD or $\overline{B}C$.
- ✓ ***Square 14*** can be covered with either prime implicants AD or $A\overline{B}$.
- ✓ ***Square 15*** can be covered with any one of four prime implicants.
  - ***Final solution:*** two essential PI with any two prime implicants that cover minterms ***3, 14, 15***: four possible ways:

1) $F = BD + \overline{B}\overline{D} + CD + AD$
2) $F = BD + \overline{B}\overline{D} + CD + A\overline{B}$
3) $F = BD + \overline{B}\overline{D} + \overline{B}C + AD$
4) $F = BD + \overline{B}\overline{D} + \overline{B}C + A\overline{B}$

3. "Don't care" conditions

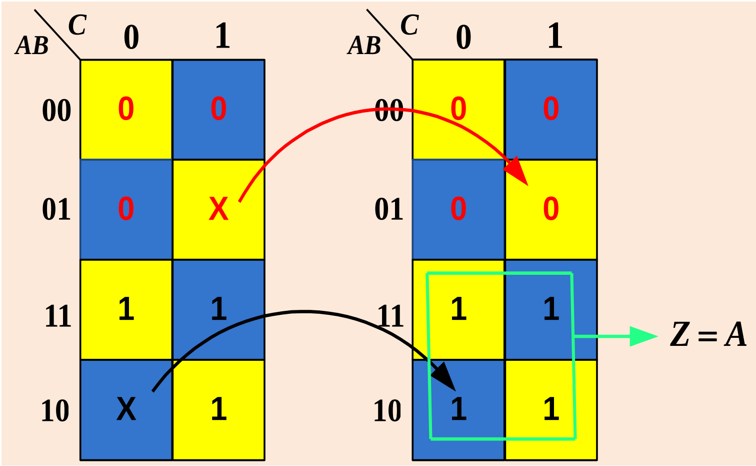
- Some logic circuits can be designed so that there are ***certain input conditions for which there are no specified output levels*** (can't happened).
- A circuit designer is ***free*** to make the output for any "***don't care condition***" either a ***0*** or a ***1*** in order to produce the simplest output.

***Example 1:* For the following truth table, use K-map to minimize the function Z**

inputs			Output
A	B	C	$Z(x,y,z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	X
1	0	0	X
1	0	1	1
1	1	0	1
1	1	1	1

Don't Care Condition

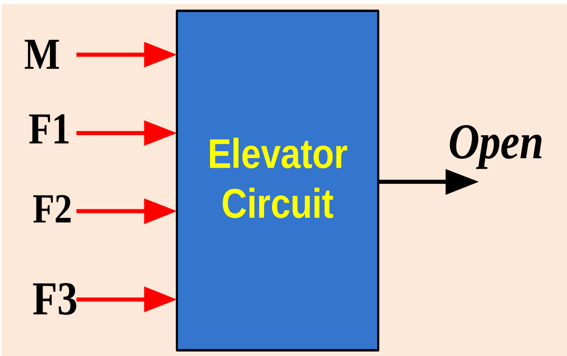
Solution:



Example 2: Design a logic circuit that controls an elevator door in a three-story building.

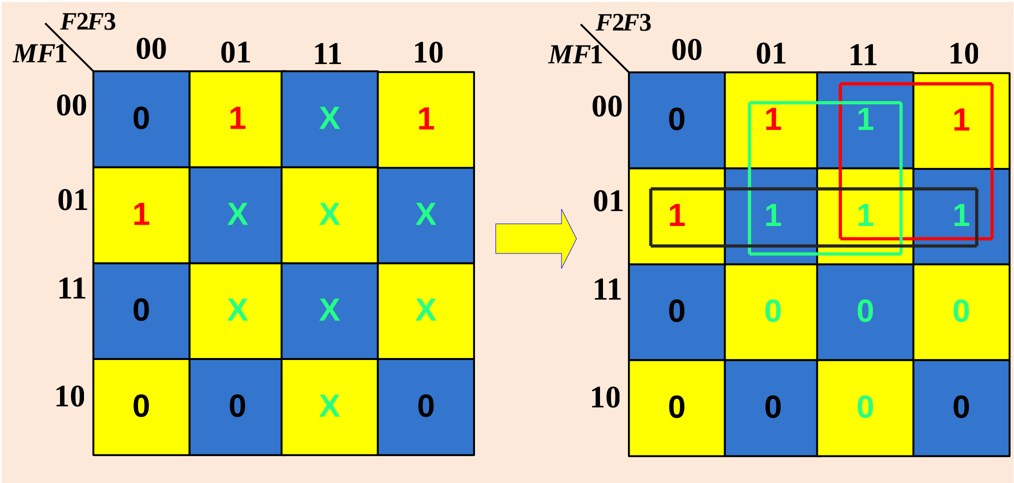
Solution:

M: ***Moving signal***: ($M = 1$: moving), ($M = 0$: stopped), $F1, F2, F3$: ***Floor indicator signals***.



Truth Table

M	*F1*	*F2*	*F3*	Open
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	X
0	1	0	0	1
0	1	0	1	X
0	1	1	0	X
0	1	1	1	X
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	X
1	1	0	0	0
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

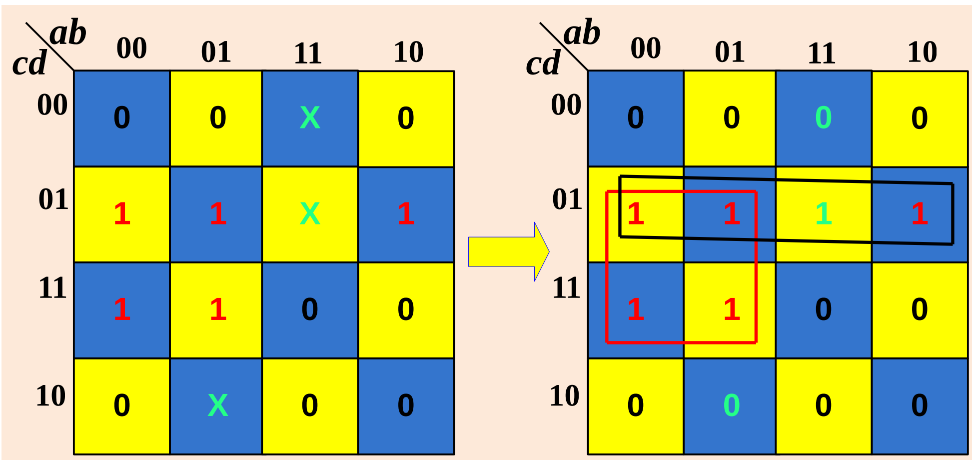


$$Open = \overline{M}(F1 + F2 + F3)$$

Example 3: ***For the following logical expression, use K-map to minimize the function F.***

$$F = \sum m(1, 3, 5, 7, 9) + \sum d(6, 12, 13)$$

Solution:

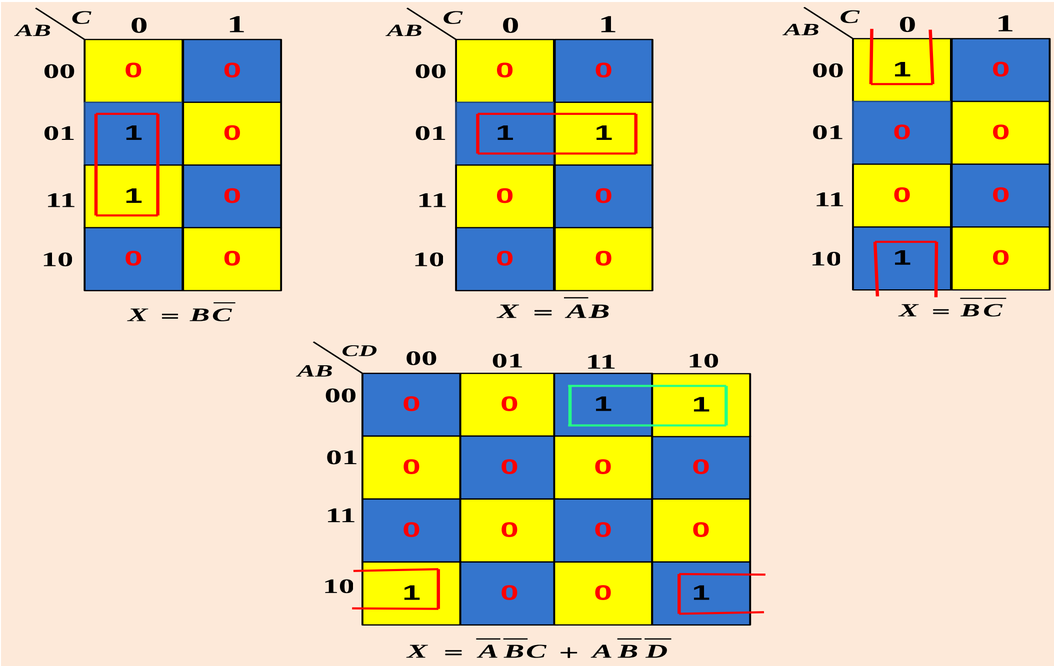


$$F = \sum m(1, 3, 5, 7, 9) + \sum d(6, 12, 13) = \bar{a}d + \bar{c}d$$

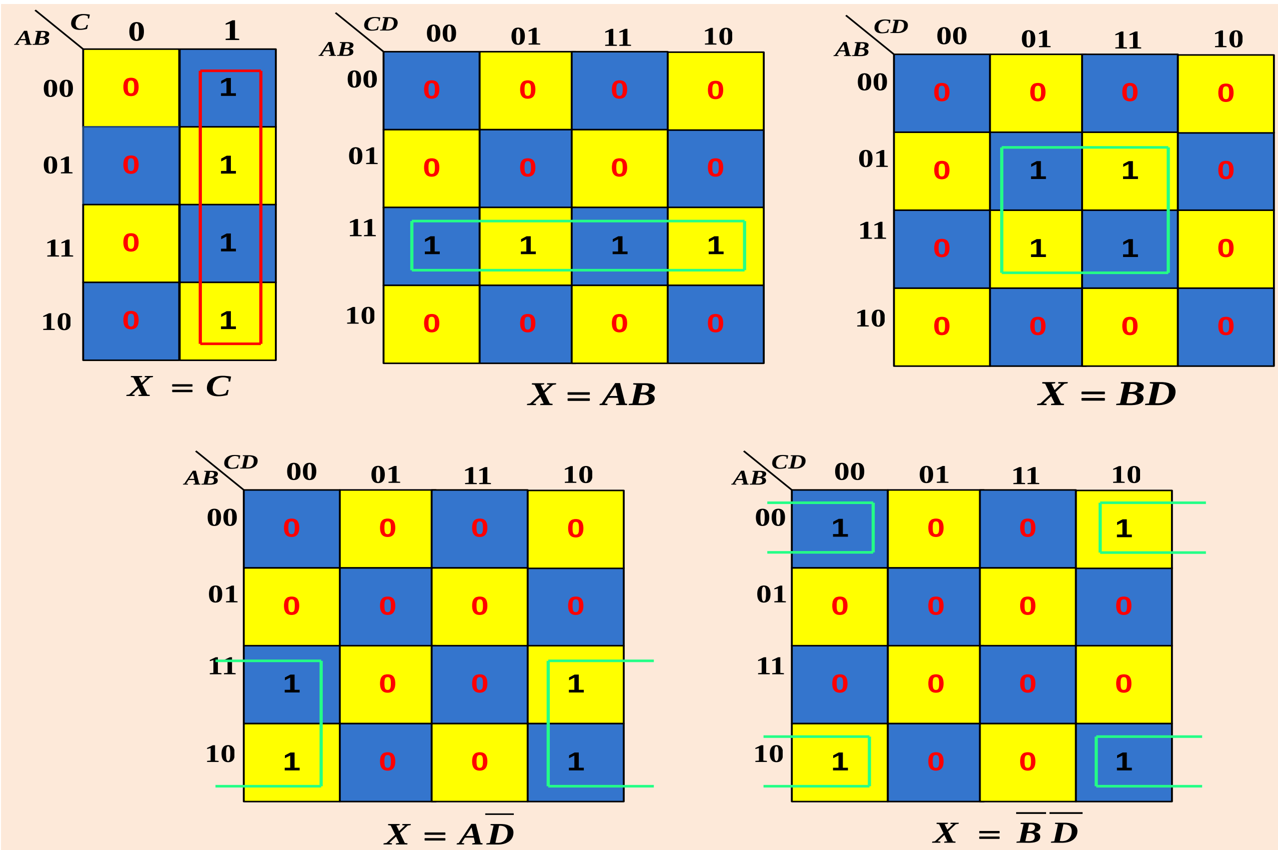
4. Summary

- Looping a pair of adjacent 1's in a k-map eliminate the variable that appears in complemented and uncomplemented form.
- Looping a quad (4) of adjacent 1's eliminates the two variables that appear in complemented and uncomplemented form.
- Looping an octet (8) of adjacent 1's eliminates the three variables that appear in complemented and uncomplemented form.

- *Looping groups of two pairs:*



- *Looping groups of four (quads):*



- *Looping groups of eight (octet):*